One-Round Communication Efficient Distributed M-Estimation

Yajie Bao
Shanghai Jiao Tong University
baoyajie2019stat@sjtu.edu.cn

Weijia Xiong
Columbia University
wjxiong5633@gmail.com

Abstract

Communication cost and local computation complexity are two main bottlenecks of the distributed statistical learning. In this paper, we consider the distributed M-estimation problem in both regular and sparse case and propose a novel one-round communication efficient algorithm. For regular distributed M-estimator, the asymptotic normality is provided to conduct statistical inference. For sparse distributed M-estimator, we only require solving a quadratic Lasso problem in the master machine using the same local information as the regular distributed M-estimator. Consequently, the computation complexity of the local machine is sufficiently reduced compared with the existing debiased sparse estimator. Under mild conditions, the theoretical results guarantee that our proposed distributed estimators achieve (near)optimal statistical convergence rate. The effectiveness of our proposed algorithm is verified through experiments across different M-estimation problems using both synthetic and real benchmark datasets.

1 Introduction

Datasets with unprecedented size are widely collected in many contemporary applications, such as social media, mobile APP, precision medicine. It also brings about tremendous challenges on computation and storage for statistical learning and estimation. Consequently, it is urgent to develop efficient distributed statistical learning and estimation methodologies. The main bottlenecks of distributed learning are communication cost and local computation complexity. The communication between local machines and the master machine is limited by the bandwidth and the computation power of local machines may be much weaker than the master machine. Therefore, efficient distributed algorithms should reduce communication rounds and local computation complexity.

This paper investigates a general M-estimation problem in the distributed environment. Let $l(\boldsymbol{X}, \cdot) : \mathbb{R}^p \to \mathbb{R}$ be the twice differentiable convex loss function and $\boldsymbol{X} \in \mathbb{R}^p$ is the random variable from some unknown probability distribution $\mathcal{D}$. The "true parameter" is defined as the minimizer of the population risk

$$\boldsymbol{\theta}^* := \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^p} \mathbb{E}_{\boldsymbol{X}\sim\mathcal{D}}[l(\boldsymbol{X}, \boldsymbol{\theta})]. \tag{1.1}$$

In practice, we estimate the true parameter $\boldsymbol{\theta}^*$ through empirical risk minimization. Now suppose we obtain a group of independent samples $\{\boldsymbol{X}_i : i = 1, 2, ..., N\}$ from the distribution $\mathcal{D}$, then M-estimator is defined as the minimizer of the empirical risk, that is

$$\widehat{\boldsymbol{\theta}} := \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^p} \frac{1}{N}\sum_{i=1}^{N} l(\boldsymbol{X}_i, \boldsymbol{\theta}). \tag{1.2}$$

Here we assume the dimensionality p can go to infinity as the sample size grows but p is less than the sample size.

With the growing dimensionality, the regular M-estimator (1.2) may suffer performance loss due to over-fitting. In addition, the model interpretation also relies on variable selection heavily. Therefore, sparse M-estimator is widely used in many real applications, which can be estimated by solving the following penalized empirical risk minimization problem:

$$\widehat{\boldsymbol{\theta}} := \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^p} \frac{1}{N}\sum_{i=1}^{N} l(\boldsymbol{X}_i, \boldsymbol{\theta}) + P_{\lambda_n}(\boldsymbol{\theta}) \tag{1.3}$$

where $P_{\lambda_n}(\boldsymbol{\theta})$ is a penalty function, such as Lasso penalty in Tishbirani (1996), Smoothing Clipped Absolute Deviation (SCAD) penalty in Fan and Li (2001), Minimax Concave penalty (MCP) in Zhang (2010).

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

In the common distributed setting, the overall sample is stored in the master machine and m local machines. Each local machine only has access to communicate with the master machine. Under this distributed environment, we focus on regular and sparse M-estimation problem with the diverging dimensionality. It is more challenging for distributed sparse M-estimation problem since (1.3) can be nonsmooth or nonconvex. There are mainly two popular distributed sparse estimators: the average debiased estimator (Lee et al., 2017; Battey et al., 2018) and multi-round estimator (Jordan et al., 2019; Chen et al., 2020a). The first estimator only require one round of communication but the computation complexity for local machines is huge. The second estimator require less local computation complexity but the number of communication rounds is not stable, which may leads great communication cost. To address the challenge of both communication cost and local computation complexity, the main goal of our paper is twofold:

1. Develop a communication efficient distributed M-estimation algorithm with one round of communication. And the local machines only need to transmit p-vectors to the master machine.

2. The sparse distributed M-estimator only require $O(Np/m)$ local computation complexity.

1.1 Related Work

Distributed statistical learning has received considerable attention in the recent years and many important topics are covered, including M-estimation (Lin and Xi, 2011; Zhang et al., 2013; Shamir et al., 2014; Wang et al., 2017b; Jordan et al., 2019; Fan et al., 2019a), high-dimensional sparse linear regression and generalized linear model (Lee et al., 2017; Battey et al., 2018), quantile regression (Chen et al., 2019, 2020b; Chen and Zhou, 2020), principal component analysis (Fan et al., 2019b; Garber et al., 2017), non-parametric regression (Zhang et al., 2015; Shang and Cheng, 2017; Han et al., 2018), support vector machine (Lian and Fan, 2018; Wang et al., 2019), deep learning (Dean et al., 2012; Anil et al., 2018; Assran et al., 2019).

The divide-and-conquer strategy is advocated by Mcdonald et al. (2009), which is simple to program and only requires a single round of communication. First each machine obtains a local estimator $\widehat{\boldsymbol{\theta}}_k$ parallelly by minimizing the average loss function based on the local samples. Then each local machine sends the local estimator to the master machine or the server. At last, the master machine takes average of all local estimators to form an aggregated estimator. Zhang et al. (2013) investigated the theoretical properties of naive average distributed M-estimator for twice differentiable loss function and obtained $O(1/\sqrt{N} + 1/n)$ convergence rate of the mean squared error, where N is the global sample size and n is the local sample size. Rosenblatt and Nadler (2016) derived the asymptotic normality of naive average distributed M-estimator under fixed dimensionality and diverging dimensionality respectively. For penalized M-estimation problem, plenty of researches (Lee et al., 2017; Battey et al., 2018) proposed to replace the local sparse penalized estimator with the debiased estimator introduced in Van de Geer et al. (2014); Javanmard and Montanari (2014), then conduct hard thresholding operator on the average debiased estimator to obtain sparsity. However, the debiased operation requires to estimate the inverse of Hessian matrix, and the computation cost is highly expensive for local machines. To achieve optimal statistical convergence rate, these divide-and-conquer estimators require the constraint of the number of local machines (Zhang et al., 2013; Lee et al., 2017; Battey et al., 2018).

To avoid the constraint on the number of local machines, the multi-round distributed estimation algorithms were developed (Zinkevich et al., 2010; Shamir et al., 2014; Wang et al., 2017a; Jordan et al., 2019). Shamir et al. (2014) proposed a distributed approximate Newton algorithm, where each local machine minimizes a modified loss function based on the gradient information from other local machines in each iteration. Jordan et al. (2019) proposed the Communication-efficient Surrogate Likelihood (CSL) framework, which approximated the global loss function by replacing the higher-order derivatives of the global loss function with the local derivatives then minimized the approximate loss function in the master machine to obtain the estimator. The corresponding sparse CSL estimator was also proposed in Jordan et al. (2019).

1.2 Our Contributions

Motivated by the aggregated estimating equation (AEE) estimator proposed in Lin and Xi (2011), this paper proposes a novel communication efficient aggregated score equation (CASE) estimator and its penalized sparse version (Pen-CASE). Different from the aforementioned average debiased estimator, our proposed Pen-CASE estimator does not require estimating the inverse Hessian matrix or exactly solving the penalized optimization problem (1.3). Both CASE and Pen-CASE achieve the goals mentioned above. Formally, our proposed method has the following contributions:

1. Pen-CASE estimator shares the same communication cost with the debiased estimator in Lee et al. (2017), while the computation complexity

of Pen-CASE estimator in each local machine is $O(np)$, which is sufficiently reduced compared with $O(np^2+p^3)$ complexity of the debiased estimator.

2. Our theoretical results show that the statistical convergence rate of the CASE estimator achieves optimal order $O_{\mathbb{P}}(\sqrt{p/N})$ under mild constraint on the number of local machines. Moreover, the Pen-CASE estimator achieves near oracle convergence rate under the sparsity assumption.

Notation

The following notations will be used throughout the paper. For a vector $\boldsymbol{x} \in \mathbb{R}^p$, ℓ_1, ℓ_2 and ℓ_∞ norm are denoted by $\|\boldsymbol{x}\|_1 = \sum_{j=1}^p |x_j|$, $\|\boldsymbol{x}\|_2 = (\sum_{j=1}^p x_j^2)^{1/2}$ and $\|\boldsymbol{x}\|_\infty = \max_j |x_j|$ respectively. For a matrix $\boldsymbol{A} = (A_{i,j}) \in \mathbb{R}^{p\times p}$, spectral norm is defined by $\|\boldsymbol{A}\|_2 = \sup_{\|\boldsymbol{x}\|_2=1} \|\boldsymbol{A}\boldsymbol{x}\|_2$, ℓ_∞ norm is defined by $\|\boldsymbol{A}\|_\infty = \max_i \sum_{j=1}^p |A_{i,j}|$. For a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and a vector $\boldsymbol{x} \in \mathbb{R}^m$: $\boldsymbol{A}_{ST}$ denotes the submatrix $(a_{s_i t_j})$ for $S = \{s_1, \ldots, s_r\} \subseteq \{1, \ldots, m\}$ and $T = \{t_1, \ldots, t_q\} \subseteq \{1, \ldots, n\}$; $\boldsymbol{x}_S$ denotes the subvector (x_j) for $j \in S$. For two sequences of positive numbers c_n and d_n, we write $c_n \lesssim d_n$ if there exists some positive constant c such that $c_n \le c d_n$ holds for sufficiently large n. For a sequence of random variables X_n, we write $X_n = O_{\mathbb{P}}(d_n)$ if for any $\varepsilon > 0$ there exists some positive constant C such that $\mathbb{P}(|X_n| > C d_n) < \varepsilon$.

2 Distributed M-Estimator

Without loss of generality, we assume all samples $\{\boldsymbol{X}_i : i = 1, 2, ..., N\}$ are stored in the master machine and m local machines evenly. Denote the data index set in the k-th machine by $\mathcal{H}_k$ with $|\mathcal{H}_k| = n$ for $k = 0, 1, ..., m$. Then the samples stored in the k-th machine are $\{\boldsymbol{X}_i : i \in \mathcal{H}_k\}$. The local empirical risk function for the k-th machine is $L_k(\boldsymbol{\theta}) = \sum_{i\in\mathcal{H}_k} l(\boldsymbol{X}_i, \boldsymbol{\theta})/n$. Suppose the dimensionality p is less than the local sample size n, it implies that there exists an unique local estimator $\widehat{\boldsymbol{\theta}}_k$ satisfying $\nabla L_k(\widehat{\boldsymbol{\theta}}_k) = 0$. Our goal is to minimize the global empirical risk, that is

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^p} \frac{1}{m+1} \sum_{k=0}^m L_k(\boldsymbol{\theta}).$$

Let $\widehat{\boldsymbol{\theta}}$ be the global M-estimator, then the score equation must satisfy

$$\frac{1}{m+1} \sum_{k=0}^m \nabla L_k(\widehat{\boldsymbol{\theta}}) = 0. \tag{2.1}$$

Denote the local Hessian matrix by $\mathbf{H}_k(\boldsymbol{\theta}) := \nabla^2 L_k(\boldsymbol{\theta})$. Taking Taylor's expansion to $\nabla L_k(\widehat{\boldsymbol{\theta}})$ around $\widehat{\boldsymbol{\theta}}_k$, we have

$$\nabla L_k(\widehat{\boldsymbol{\theta}}) = \mathbf{H}_k(\widetilde{\boldsymbol{\theta}}_k) \left(\widehat{\boldsymbol{\theta}} - \widehat{\boldsymbol{\theta}}_k\right) \tag{2.2}$$

where $\widetilde{\boldsymbol{\theta}}_k$ lies between $\boldsymbol{\theta}$ and $\widehat{\boldsymbol{\theta}}_k$. Then substitute (2.2) to (2.1) we have

$$\frac{1}{m+1} \sum_{k=0}^m \mathbf{H}_k(\widetilde{\boldsymbol{\theta}}_k) \left(\widehat{\boldsymbol{\theta}} - \widehat{\boldsymbol{\theta}}_k\right) = 0. \tag{2.3}$$

We replace $\mathbf{H}_k(\widetilde{\boldsymbol{\theta}}_k)$ by $\mathbf{H}_k(\widehat{\boldsymbol{\theta}}_k)$ since $\widetilde{\boldsymbol{\theta}}_k$ is unknown and closed to $\widehat{\boldsymbol{\theta}}_k$, which leads the following AEE estimator proposed in Lin and Xi (2011)

$$\widehat{\boldsymbol{\theta}}_{\text{AEE}} := \left(\sum_{k=0}^m \mathbf{H}_k(\widehat{\boldsymbol{\theta}}_k)\right)^{-1} \sum_{k=0}^m \mathbf{H}_k(\widehat{\boldsymbol{\theta}}_k)\widehat{\boldsymbol{\theta}}_k. \tag{2.4}$$

However, Lin and Xi (2011) required the loss function $l(\boldsymbol{X}_i, \boldsymbol{\theta})$ is three-times differentiable. Here we generalize it to twice differentiable loss functions, which can be applied in broader distributed M-estimation problems.

To obtain AEE estimator (2.4), the local machines need to transmit $\mathbf{H}_k(\widehat{\boldsymbol{\theta}}_k)$ and $\mathbf{H}_k(\widehat{\boldsymbol{\theta}}_k)\widehat{\boldsymbol{\theta}}_k$ to the master machine and the communication cost is $O(p^2+p)$. Given the fact the samples in each local machine are from the same distribution, $\mathbf{H}_0(\widehat{\boldsymbol{\theta}}_k)$ should be closed to $\mathbf{H}_k(\widehat{\boldsymbol{\theta}}_k)$. We propose using $\mathbf{H}_0(\widehat{\boldsymbol{\theta}}_k)$ to replace $\mathbf{H}_k(\widehat{\boldsymbol{\theta}}_k)$, which means that each local machine only needs to transmit $\widehat{\boldsymbol{\theta}}_k$ to the master machine, then $\mathbf{H}_0(\widehat{\boldsymbol{\theta}}_k)$ can be computed using $\widehat{\boldsymbol{\theta}}_k$ and the sample in the master machine. with the help of the local Hessian replication, the communication cost is reduced to $O(p)$. We call this estimator as communication-efficient aggregate score equation (CASE) estimator, which can be written as

$$\widehat{\boldsymbol{\theta}}_{\text{CASE}} = \left(\sum_{k=0}^m \mathbf{H}_0(\widehat{\boldsymbol{\theta}}_k)\right)^{-1} \sum_{k=0}^m \mathbf{H}_0(\widehat{\boldsymbol{\theta}}_k)\widehat{\boldsymbol{\theta}}_k. \tag{2.5}$$

It's worthwhile noting that the CASE estimator satisfies that $\widehat{\boldsymbol{\theta}}_{\text{CASE}} = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^p} D_n(\boldsymbol{\theta})$ where

$$D_n(\boldsymbol{\theta}) = \frac{1}{m+1}\left\{\frac{1}{2}\boldsymbol{\theta}^{\mathrm{T}}\left(\sum_{k=0}^m \mathbf{H}_0(\widehat{\boldsymbol{\theta}}_k)\right)\boldsymbol{\theta} - \left(\sum_{k=0}^m \mathbf{H}_0(\widehat{\boldsymbol{\theta}}_k)\widehat{\boldsymbol{\theta}}_k\right)^{\mathrm{T}}\boldsymbol{\theta}\right\}.$$

To encourage sparsity, we add Lasso penalty to $D_n(\boldsymbol{\theta})$, then we can obtain the penalized CASE estimator (Pen-CASE) by solving the following quadratic Lasso problem

$$\widehat{\boldsymbol{\theta}}_{\text{Pen-CASE}} = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^p} D_n(\boldsymbol{\theta}) + \lambda_n \|\boldsymbol{\theta}\|_1, \tag{2.6}$$

where λ_n is the tuning parameter. We provide two M-estimation examples on linear regression and logistic regression to illustrate the application of our proposed CASE estimator and Pen-CASE estimator.

Example 2.1 (Linear regression)**.** *Consider the following linear regression model*

$$Y = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\theta}^* + \epsilon$$

where $\boldsymbol{\theta}^* \in \mathbb{R}^p$ is the true parameter to be estimated, $\boldsymbol{X}$ is the zero-mean covariate and ϵ is the zero-mean random noise. For the linear regression model, the local estimator is given by $\widehat{\boldsymbol{\theta}}_k = (\sum_{i\in\mathcal{H}_k}\boldsymbol{X}_i\boldsymbol{X}_i^{\mathrm{T}})^{-1}\sum_{i\in\mathcal{H}_k}\boldsymbol{X}_iY_i$. Then the CASE estimator is exactly the naive aggregate estimator

$$\widehat{\boldsymbol{\theta}}_{CASE} = \frac{1}{m+1}\sum_{k=0}^{m}\widehat{\boldsymbol{\theta}}_k.$$

And the Pen-CASE estimator is given by

$$\arg\min_{\boldsymbol{\theta}\in\mathbb{R}^p}\frac{1}{2}\boldsymbol{\theta}^{\mathrm{T}}\widehat{\mathbf{C}}_0\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{CASE}^{\mathrm{T}}\widehat{\mathbf{C}}_0\boldsymbol{\theta} + \lambda_n\|\boldsymbol{\theta}\|_1$$

where $\widehat{\mathbf{C}}_0 = \sum_{i\in\mathcal{H}_0}\boldsymbol{X}_i\boldsymbol{X}_i^{\mathrm{T}}/n$.

Example 2.2 (Logistic regression)**.** *Consider the following logistic regression model,*

$$\mathbb{P}(Y=1|\boldsymbol{X}) = 1-\mathbb{P}(Y=-1|\boldsymbol{X}) = \left(1+\exp\left(-\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\theta}^*\right)\right)^{-1}$$

where $Y \in \{-1,1\}$ is the label, $\boldsymbol{X}$ is the covariate with zero mean and $\boldsymbol{\theta}^* \in \mathbb{R}^p$ is the true parameter to be estimated. The loss function is negative likelihood function

$$l(\boldsymbol{X}_i,\boldsymbol{\theta}) = \log\left(1+\exp\left(-Y_i\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}\right)\right).$$

Let $\widehat{\boldsymbol{\theta}}_k$ be the local estimator from the k-th local machine then the local Hessian matrix is given by

$$\mathbf{H}_0(\widehat{\boldsymbol{\theta}}_k) = \sum_{i\in\mathcal{H}_0}\boldsymbol{X}_i\boldsymbol{X}_i^{\mathrm{T}}p_i(\widehat{\boldsymbol{\theta}}_k)\left(1-p_i(\widehat{\boldsymbol{\theta}}_k)\right)$$

where $p_i(\widehat{\boldsymbol{\theta}}_k) = (1+\exp(-\boldsymbol{X}_i^{\mathrm{T}}\widehat{\boldsymbol{\theta}}_k))^{-1}$.

3 Algorithmic Framework

The implementation of our proposed distributed M-estimation method is stated in Algorithm 1. First, we compute the local estimator $\widehat{\boldsymbol{\theta}}_k$ by minimizing $L_k(\boldsymbol{\theta})$ in each machine parallelly. Here we use gradient descent algorithm to obtain $\widehat{\boldsymbol{\theta}}_k$, which will not bring about great computation burden to local machines. Then each local machine transmits $\widehat{\boldsymbol{\theta}}_k$ to the master machine and $\mathbf{H}_0(\widehat{\boldsymbol{\theta}}_k)$ is obtained in the master machine. To avoid computing the inverse of matrix, we obtain the CASE estimator by solving a linear system.

In order to obtain the local estimator $\widehat{\boldsymbol{\theta}}_k$ and Hessian matrix $\mathbf{H}_0(\widehat{\boldsymbol{\theta}}_k)$, the computation complexities are respectively $O(np)$ and $O(np^2)$. For CASE estimator, the computation complexity of solving the linear system is $O(p^3)$. For Pen-CASE estimator, the quadratic Lasso problem can be solved efficiently by LARS algorithm (Efron et al., 2004) or coordinate descent algorithm and the computation complexity is also $O(p^3)$. Therefore, both CASE estimator and Pen-CASE estimator require $O(Np + np^2)$ total time complexity and $O(np)$ local time complexity. However, the local computation complexity of the average debiased estimator in Lee et al. (2017); Battey et al. (2018) is $O(np^2 + p^3)$.

Algorithm 1: One-Round Communication Efficient Distributed M-Estimation

```
Input: Data $\{\boldsymbol{X}_i : i \in \mathcal{H}_k\}$ for $k = 0, 1, ..., m$, the
  initial value $\boldsymbol{\theta}^{(0)}$, the learning rate $\eta$ and the
  number of iterations $T$ for gradient descent, the
  tuning parameter $\lambda_n$.
Output: $\widehat{\boldsymbol{\theta}}_{\text{CASE}}$ and $\widehat{\boldsymbol{\theta}}_{\text{Pen-CASE}}$.
for $k = 0, 1, ..., m$ do
    The $k$-th local machine: Set $\boldsymbol{\theta}_k^{(0)} = \boldsymbol{\theta}^{(0)}$,
    for $t = 1, .., T$ do
        Update estimator by
            $\boldsymbol{\theta}_k^{(t)} = \boldsymbol{\theta}_k^{(t-1)} - \eta\nabla L_k(\boldsymbol{\theta}_k^{(t-1)})$.
    end
    Set $\widehat{\boldsymbol{\theta}}_k = \boldsymbol{\theta}_k^{(T)}$ and send $\widehat{\boldsymbol{\theta}}_k$ to the matser
      machine.
end
The master machine: Compute the Hessian
  matrices $\mathbf{H}_0(\widehat{\boldsymbol{\theta}}_k)$ for $k = 0, 1, ..., m$, then obtain
  CASE estimator by solving the following linear
  system about $\boldsymbol{x}$
    $\left(\sum_{k=0}^{m}\mathbf{H}_0(\widehat{\boldsymbol{\theta}}_k)\right)\boldsymbol{x} = \sum_{k=0}^{m}\mathbf{H}_0(\widehat{\boldsymbol{\theta}}_k)\widehat{\boldsymbol{\theta}}_k$.
The master machine: Obtain the Pen-CASE
  estimator by
    $\widehat{\boldsymbol{\theta}}_{\text{Pen-CASE}} = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^p} D_n(\boldsymbol{\theta}) + \lambda_n\|\boldsymbol{\theta}\|_1$.    (3.1)
```

4 Theoretical Results

Denote the population Hessian matrix by $\mathbf{I}(\boldsymbol{\theta}) := \mathbb{E}(\nabla^2 l(\boldsymbol{X},\boldsymbol{\theta}))$. Let the Euclidean ball around $\boldsymbol{\theta}^*$ with radius $\rho > 0$ be $U = \{\boldsymbol{\theta} \in \mathbb{R}^p : \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2 \leq \rho\}$.

We impose the following regular conditions to help establish the theoretical results for our proposed CASE estimator and Pen-CASE estimator.

(**C**1) The parameter space $\boldsymbol{\Theta} \subseteq \mathbb{R}^p$ is compact and convex. $\boldsymbol{\theta}^*$ is an interior point in $\boldsymbol{\Theta}$ and the ℓ_2-radius of parameter space $R := \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2 > 0$.

(**C**2) For any $\varepsilon > 0$, there exists some $\delta > 0$ such that

$$\liminf_{n\to\infty} \mathbb{P}\left(\inf_{\|\boldsymbol{\theta}-\boldsymbol{\theta}^*\|_2\geq\delta}[l(\boldsymbol{X},\boldsymbol{\theta}) - l(\boldsymbol{X},\boldsymbol{\theta}^*)] \geq \varepsilon\right) = 1.$$

(**C**3) The population risk is twice differentiable and the smallest and largest eigenvalue of $\mathbf{I}(\boldsymbol{\theta}^*)$ satisfy that $\lambda_- \leq \lambda_{\min}(\mathbf{I}(\boldsymbol{\theta}^*)) \leq \lambda_{\min}(\mathbf{I}(\boldsymbol{\theta}^*)) \leq \lambda_+$ for two positive constants λ_- and λ_+.

(**C**4) The sample risk is twice differentiable and there exist some positive constant L and integer $K \geq 2$ such that for all $\boldsymbol{\theta} \in U$

$$\mathbb{E}\left(\left\|\nabla^2 l(\boldsymbol{X},\boldsymbol{\theta}) - \mathbf{I}(\boldsymbol{\theta})\right\|_2^K\right) \leq L^K.$$

Moreover for all $\boldsymbol{\theta}, \boldsymbol{\theta}^{'} \in U$,

$$\left\|\nabla^2 l(\boldsymbol{X},\boldsymbol{\theta}) - \nabla^2 l(\boldsymbol{X},\boldsymbol{\theta}^{'})\right\|_2 \leq M(\boldsymbol{X})\left\|\boldsymbol{\theta} - \boldsymbol{\theta}^{'}\right\|_2$$

where $\mathbb{E}(M^K(\boldsymbol{X})) \leq M^K$ for some positive constant M.

(**C**5) The sample gradient evaluated at $\boldsymbol{\theta}^*$ satisfies the sub-gaussian property: for any $\lambda \in \mathbb{R}$ such that

$$\sup_{\|\boldsymbol{u}\|_2=1} \mathbb{E}\left[\exp\left(\lambda|\boldsymbol{u}^{\mathrm{T}}\nabla l(\boldsymbol{X},\boldsymbol{\theta}^*)|\right)\right] \leq \exp\left(\lambda^2\right).$$

Condition (**C**1) and (**C**2) are very common regular conditions in M-estimation problem. Condition (**C**3) requires the population risk is strongly convex on $\boldsymbol{\theta}^*$. Combining condition (**C**3) and (**C**4), the strong convexity of the sample risk function holds locally around $\boldsymbol{\theta}^*$ with high probability. Condition (**C**1)-(**C**4) also appear in Zhang et al. (2013); Jordan et al. (2019). Condition (**C**5) is used to establish the upper bound for the empirical process in M-estimation.

To begin with, we define the following error term

$$B_k = \frac{M\|\nabla L_k(\boldsymbol{\theta}^*)\|_2}{(1-\rho)^2\lambda_-^2} + \frac{\|\mathbf{H}_k(\boldsymbol{\theta}^*) - \mathbf{H}_0(\boldsymbol{\theta}^*)\|_2}{(1-\rho)\lambda_-} \tag{4.1}$$

for $k = 0, 1, ..., m$. In fact, (4.1) represents the error brought by the operation that we replace $\mathbf{H}_k(\widehat{\boldsymbol{\theta}}_k)$ with $\mathbf{H}_0(\widehat{\boldsymbol{\theta}}_k)$.

4.1 CASE Estimator

The following theorem provides the Bahadur representation for our proposed CASE estimator. The ℓ_2 error bound and asymptotic normality can be easily obtained from Theorem 4.1.

Theorem 4.1. *Under condition* (**C**1)*-*(**C**5)*, for CASE estimator* (2.5) *we have*

$$\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_{CASE} = \mathbf{I}(\boldsymbol{\theta}^*)^{-1}\frac{1}{N}\sum_{k=0}^{m}\sum_{i\in\mathcal{H}_k}\nabla l(\boldsymbol{X}_i,\boldsymbol{\theta}^*) + \boldsymbol{u}_n, \tag{4.2}$$

where

$$\|\boldsymbol{u}_n\|_2 \leq C\max_{0\leq k\leq m}\|\nabla L_k(\boldsymbol{\theta}^*)\|_2 B_k$$

with probability at least

$$1 - m\left(2e^{(-c_0 n + 2p)} + c_1 n^{-K/2} + c_2\left(\frac{\log 2p}{n}\right)^{K/2}\right),$$

for four positive constants C, c_0, c_1 and c_2.

The first term in the right hand side of (4.2) is empirical process of the Bahadur representation in the centralized sample case. From Lemma A.1 in the Appendix, we have $\max_{0\leq k\leq m}\|\nabla L_k(\boldsymbol{\theta}^*)\|_2 = O_{\mathbb{P}}(\sqrt{p/n})$, $\|\sum_{k=0}^{m}\nabla L_k(\boldsymbol{\theta}^*)/(m+1)\|_2 = O_{\mathbb{P}}(\sqrt{p/N})$ and $\max_{0\leq k\leq m}\|\mathbf{H}_k(\boldsymbol{\theta}^*) - \mathbf{H}_0(\boldsymbol{\theta}^*)\|_2 = O_{\mathbb{P}}(\sqrt{\log p/n})$. Then Theorem 4.1 also indicates that the convergence rate of the ℓ_2 estimation error for CASE estimator is

$$\left\|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_{\mathrm{CASE}}\right\|_2 = O_{\mathbb{P}}\left(\sqrt{\frac{p}{N}} + \frac{p}{n}\right).$$

If the number of local machines satisfies $m \lesssim \sqrt{N/p}$, the CASE estimator achieves optimal statistical convergence rate.

From Theorem 4.1, we can easily obtain the asymptotic normality of CASE estimator as following.

Corollary 4.1. *Under condition* (**C**1)*-*(**C**5)*, if the number of machines satisfies $m = o(\sqrt{N}/p)$, the asymptotic normality of CASE estimator holds,*

$$\sqrt{N}\left(\widehat{\boldsymbol{\theta}}_{CASE} - \boldsymbol{\theta}^*\right) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$$

where $\boldsymbol{\Sigma} = \mathbf{I}(\boldsymbol{\theta}^*)^{-1}\mathbb{E}(\nabla l(\boldsymbol{X},\boldsymbol{\theta}^*)\nabla l(\boldsymbol{X},\boldsymbol{\theta}^*)^{\mathrm{T}})\mathbf{I}(\boldsymbol{\theta}^*)^{-1}$.

For generalized linear model, the asymptotic covariance matrix $\boldsymbol{\Sigma}$ is the inverse of Fisher information matrix $\mathbf{I}(\boldsymbol{\theta}^*)^{-1}$. To conduct statistical inference, such as hypothesis test and confidence interval, we need to estimate the asymptotic covariance matrix in the distributed environment. To prevent transmitting matrices, we provide a communication efficient plug-in covariance estimator as

$$\widehat{\boldsymbol{\Sigma}} = \frac{n}{m+1}\mathbf{H}_0(\widehat{\boldsymbol{\theta}}_{\mathrm{CASE}})^{-1}\boldsymbol{G}\mathbf{H}_0(\widehat{\boldsymbol{\theta}}_{\mathrm{CASE}})^{-1} \tag{4.3}$$

where $\boldsymbol{G} = \sum_{k=0}^{m}\nabla L_k(\widehat{\boldsymbol{\theta}}_{\mathrm{CASE}})\nabla L_k(\widehat{\boldsymbol{\theta}}_{\mathrm{CASE}})^{\mathrm{T}}$. Obviously, $\widehat{\boldsymbol{\Sigma}}$ only requires that the local machines send gradient vector $\nabla L_k(\widehat{\boldsymbol{\theta}}_{\mathrm{CASE}})$ to the master machine.

Lemma A.3 in the Appendix indicates that the plug-in covariance estimator $\|\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}\|_2 \to 0$ with probability tending to 1. In fact, the spectral norm bound is dominated by $\|\mathbf{H}_0(\widehat{\boldsymbol{\theta}}_{\text{CASE}})^{-1} - \mathbf{I}(\boldsymbol{\theta}^*)^{-1}\|_2$. In practice, the sample size of the matser machine can be much larger than that of the local machine, which means that we can obtain more accurate covariance matrix estimation.

The CASE estimator for the linear regression model is just the naive aggregate estimator, which is investigated in Zhang et al. (2013). The following theorem provides the ℓ_2 error bound of CASE estimator for the logistic regression model.

Theorem 4.2. *For the logistic regression model in Example 2.2, assume the covariate $\boldsymbol{X}$ satisfies that $\sup_{\|\boldsymbol{u}\|_2=1} \mathbb{E}[\exp(\lambda|\boldsymbol{u}^{\mathrm{T}}\boldsymbol{X}|)] \le \exp(\lambda^2)$ for any $\lambda \in \mathbb{R}$. Moreover, assume $\lambda_{\min}(\mathbb{E}[\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}]) \ge C_1$ and $\|\boldsymbol{\theta}^*\|_2 \le C_2$ for two positive constants C_1 and C_2. Then the CASE estimator satisfies that*

$$\|\widehat{\boldsymbol{\theta}}_{CASE} - \boldsymbol{\theta}^*\|_2 = O_{\mathbb{P}}\left(\sqrt{\frac{p}{N}} + \frac{p}{n}\right).$$

4.2 Pen-CASE Estimator

In this section, we will establish the estimation error bound and support recovery for the Pen-CASE estimator. Before that, we need to introduce the following additional conditions for Pen-CASE estimator.

(**C**6) The support set of $\boldsymbol{\theta}^*$ is $S := \{j : |\theta_j^*| \neq 0\}$ and the cardinality of S is s.

(**C**7) The population Hessian matrix evaluated at $\boldsymbol{\theta}^*$ satisfies that

$$\left\|\mathbf{I}(\boldsymbol{\theta}^*)_{S^cS}\mathbf{I}(\boldsymbol{\theta}^*)_{SS}^{-1}\right\|_\infty \le \alpha \qquad (4.4)$$

for $0 < \alpha < 1$.

The incoherence condition (**C**7) is necessary for the variable selection consistency of Lasso penalized estimator.

Theorem 4.3. *Under condition (**C**1)-(**C**6) and set*

$$\lambda_n \ge C_3 \left\{ \left\| \frac{1}{m+1} \sum_{k=0}^{m} \nabla L_k(\boldsymbol{\theta}^*) \right\|_\infty + \max_{0 \le k \le m} \|\nabla L_k(\boldsymbol{\theta}^*)\|_2 B_k \right\}$$

for some sufficiently large positive constant C_3, then we have

$$\left\|\widehat{\boldsymbol{\theta}}_{Pen\text{-}CASE} - \boldsymbol{\theta}^*\right\|_2 \le \frac{6\sqrt{s}}{(1-\rho)\lambda_-}\lambda_n \qquad (4.5)$$

with probabilty at least

$$1 - m\left(2e^{(-c_0 n + 2p)} + c_1 n^{-K/2} + c_2 \left(\frac{\log 2p}{n}\right)^{K/2}\right).$$

From the proof of Theorem 4.3, we can obtain $\|\sum_{k=0}^m \nabla L_k(\boldsymbol{\theta}^*)/(m+1)\|_\infty = O_{\mathbb{P}}(\sqrt{\log N/N})$. It yields the ℓ_2 estimation error of the Pen-CASE estimator

$$\left\|\widehat{\boldsymbol{\theta}}_{\text{Pen-CASE}} - \boldsymbol{\theta}^*\right\|_2 = O_{\mathbb{P}}\left(\sqrt{\frac{s\log N}{N}} + \frac{p\sqrt{s}}{n}\right).$$

If the number of local machines satisfies $m \lesssim \sqrt{N\log N}/p$, combing with the convergence rate of $\max_{0\le k\le m} B_k$, we can obtain

$$\left\|\widehat{\boldsymbol{\theta}}_{\text{Pen-CASE}} - \boldsymbol{\theta}^*\right\|_2 = O_{\mathbb{P}}\left(\sqrt{\frac{s\log N}{N}}\right). \qquad (4.6)$$

In fact, the constraint on the number of local machines is not rigorous since the dimensionality p is much smaller than the total sample size N. It's worthwhile noting that the ℓ_2 error bound of the Pen-CASE estimator (4.6) nearly matches the oracle convergence rate $O_{\mathbb{P}}(\sqrt{s/N})$ in the centralized sample case.

Denote the support set of the Pen-CASE estimator by $\widehat{S} := \{j : |\widehat{\theta}_j| \neq 0\}$. The following theorem provides ℓ_∞ estimation error bound for the Pen-CASE estimator.

Theorem 4.4. *Under condition (**C**1)-(**C**7) and suppose $m \lesssim \sqrt{N\log N}/p$, with the same choice for λ_n in Theorem 4.3, we have $\widehat{S} \subseteq S$ holds with probability tending to 1. Moreover, the ℓ_∞ estimation error bound is given as*

$$\left\|\widehat{\boldsymbol{\theta}}_{Pen\text{-}CASE} - \boldsymbol{\theta}^*\right\|_\infty = O_{\mathbb{P}}\left(\|\mathbf{I}(\boldsymbol{\theta}^*)_{SS}^{-1}\|_\infty \sqrt{\frac{\log N}{N}}\right). \quad (4.7)$$

According to (4.7), if $\boldsymbol{\theta}^*$ satisfies that

$$\min_{j\in S}|\theta_j^*| \ge C_4 \|\mathbf{I}(\boldsymbol{\theta}^*)_{SS}^{-1}\|_\infty \sqrt{\frac{\log N}{N}} \qquad (4.8)$$

for some sufficiently large positive constant C_4, then we have $\widehat{S} = S$ with probability tending to 1. It means that our proposed Pen-CASE estimator can achieve exact support recovery for general sparse M-estimation problem in moderately high dimension.

The following theorem provides estimation error bound for the Pen-CASE estimator in the sparse linear regression.

Theorem 4.5. *For the linear regression model in Example 2.1, assume each coordinate of covariate $\boldsymbol{X}$ and the random noise ϵ are both subgaussian random variables with parameter 1. Let $\mathbf{C} := \mathbb{E}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})$, assume $\lambda_- \le \lambda_{\min}(\mathbf{C}) \le \lambda_{\min}(\mathbf{C}) \le \lambda_+$. Suppose the support set of $\boldsymbol{\theta}^*$ is S with sparsity s. If the number of local machines satisfies $m \lesssim \sqrt{N\log N}/p$ and set*

$$\lambda_n \ge C_5\sqrt{\frac{\log N}{N}}$$

for some positive constant C_5, we have

1. *ℓ_2 error:*

$$\left\|\widehat{\boldsymbol{\theta}}_{Pen\text{-}CASE}-\boldsymbol{\theta}^*\right\|_2=O_{\mathbb{P}}\left(\sqrt{\frac{s\log N}{N}}\right);$$

2. *ℓ_∞ error:*

$$\left\|\widehat{\boldsymbol{\theta}}_{Pen\text{-}CASE}-\boldsymbol{\theta}^*\right\|_\infty=O_{\mathbb{P}}\left(\|\mathbf{C}_{SS}^{-1}\|_\infty\sqrt{\frac{\log N}{N}}\right).$$

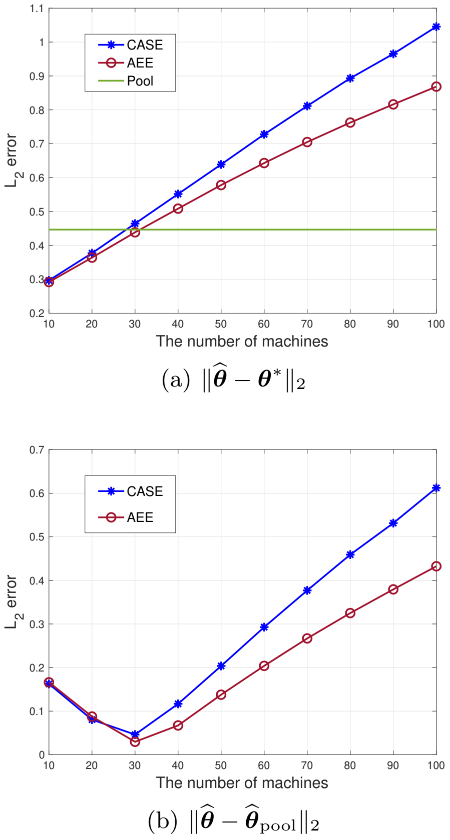
The convergence rates of estimation errors for Pen-CASE estimator in sparse linear regression model match the optimal rates in Wainwright (2009).

5 Experiments

In this section, we conduct experiments to show the effectiveness of the proposed CASE and Pen-CASE estimator using both synthetic and real datasets.

5.1 Synthetic Data

Regular logistic regression model. In the first simulation, we compare CASE estimator and AEE estimator in the regular logistic regression model using synthetic data. For the logistic regression model in Example 2.2, we independently generate the covariate $\boldsymbol{X}_i$ from the multivariate normal distribution $\mathcal{N}(\mathbf{0},\mathbf{I}_p)$, where $\mathbf{I}_p$ is a p dimensional identity matrix. Each entry of $\boldsymbol{\theta}^*$ follows i.i.d uniform distribution over the interval $(-1,1)$. Each observation Y_i is generated independently from the binomial distribution with parameter $p_i=(1+\exp(-\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}^*))^{-1}$. Here we fix the dimension $p=20$ and the size of training set $N=10{,}000$. We use ℓ_2 estimation error $\|\widehat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*\|_2$ and $\|\widehat{\boldsymbol{\theta}}-\widehat{\boldsymbol{\theta}}_{\text{pool}}\|_2$ to evaluate different estimators, where $\widehat{\boldsymbol{\theta}}_{\text{pool}}$ is the centralized estimator using the whole training set. The experiment is repeated 50 times and the averages of results are demonstrated in Figure 1. From Figure 1(a), we can see that the ℓ_2 estimation error of AEE estimator is lower than CASE estimator because we replace $\mathbf{H}_k(\widehat{\boldsymbol{\theta}}_k)$ with $\mathbf{H}_0(\widehat{\boldsymbol{\theta}}_k)$. In addition, the performances of AEE estimator and CASE estimator are better than the centralized estimator when the number of machines is less than 30. The reason is that the dominant term of error bound in Theorem 4.2 is p/n when the number of machines is small. Figure 1(b) illustrates the performance loss of each estimator due to the distributed estimation, which increases as the local sample size decreases.



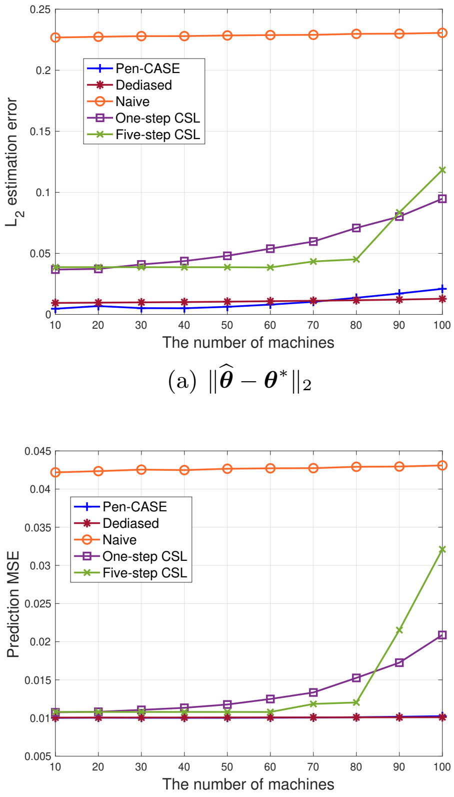
(a) $\|\widehat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*\|_2$

(b) $\|\widehat{\boldsymbol{\theta}}-\widehat{\boldsymbol{\theta}}_{\text{pool}}\|_2$

Figure 1: The ℓ_2 errors of CASE estimator and AEE estimator in the regular logistic regression versus the number of machines (including the master machine).

Sparse linear regression model. In the second simulation, we apply Pen-CASE estimator to a sparse linear regression model using synthetic data. We compare the performance of Pen-CASE estimator with the naive average sparse estimator, the average debiased estimator (Lee et al., 2017) and penalized CSL estimator (Jordan et al., 2019). For the linear regression model in Example 2.1, the covariate $\boldsymbol{X}_i$ is generated independently from the multivariate normal distribution $\mathcal{N}(\mathbf{0},\mathbf{C})$ where $\mathbf{C}_{j,k}=0.5^{|j-k|}$. We set sparsity $s=5$ in our simulation and s nonzero entries of $\boldsymbol{\theta}^*$ are generated independently from a uniform distribution over the interval $(-5,5)$. Then the observation is generated as $Y_i=\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}^*+\epsilon_i$, where the random noise ϵ_i follows i.i.d normal distribution $\mathcal{N}(0,0.1)$. We fix the sample size of training set as $N=20{,}000$ and the tuning parameter is selected using an independent validation set with size 1,000. The ℓ_2 estimation error $\|\widehat{\boldsymbol{\theta}}-\boldsymbol{\theta}^*\|_2$ and the prediction mean squared error (MSE) $\sum_{i=1}^n(\widehat{Y}_i-Y_i)^2/n$ are used to evaluate the performance of each method. The prediction MSE is computed in an independent test set with size 1,000. For penalized CSL estimator, we compute the two metrics after one and five communication rounds.

The average ℓ_2 estimation error and prediction mean squared error (MSE) over 50 trails are illustrated in



(a) $\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_2$

(b) prediction MSE

Figure 2: The ℓ_2 estimation error and prediction MSE of each estimator in the sparse linear regression versus the number of machines (including the master machine).

Figure 2. It indicates that the Pen-CASE estimator and the average debiased estimator have the best similar performance. From Figure 2(a), it can be seen that the ℓ_2 estimation error of the Pen-CASE estimator is slightly greater than the average debiased estimator when the number of machines is great than 70. The reason is that the second term in estimation error of Pen-CASE estimator is greater than that of the average debiased estimator when m is large. However, as we discussed in the section 3, the Pen-CASE estimator sufficiently reduces the computation complexity of the local machine compared with the average debiased estimator. Thus the Pen-CASE estimator is more efficient in practice. Moreover, the performance of the penalized CSL estimator becomes weak and unstable as the sample size in the master machine decreases.

5.2 Real Data

MNIST Data. In the first experiment, we apply the CASE estimator to multi-class logistic regression model using the MNIST (Lecun et al., 1998) dataset. We also compare our CASE estimator with naive average estimator, AEE estimator (Lin and Xi, 2011) and CSL estimator (Jordan et al., 2019). The total sample size of the training set and test set are respectively 60,000 and 10,000. In the beginning, we filter out features with more than 59,000 zeros on the training set to make sure that the Hessian matrix in each machine is non-singular, and the number of remaining features is 467. The training set is randomly divided into 40 machines evenly. We repeat the experiment ten times and take the average of classification errors to evaluate the performance of different estimators. The results are summarized in Table 1, which shows that AEE estimator has the best performance. The classification error of CASE estimator is slightly higher than AEE estimator due to the Hessian matrix replacement. Naive average estimator has the similar performance as the CASE estimator because the two estimator has the same convergence rate according to Zhang et al. (2013) and Theorem 4.1. The classification error of CSL estimator after 10 communication rounds is still higher than other three estimators.

Table 1: Classification errors and standard deviations of different estimators in MNIST dataset.

Estimator	Naive	CASE	AEE	5-step CSL	10-step CSL
Error	0.1312	0.1304	0.1244	0.2578	0.2173
SD	0.0002	0.0006	0.0006	0.0253	0.0068

w8a Data. In the second experiment, we use w8a datset (Platt, 1998) to verify the performance of different estimators in sparse logistic regression. The w8a dataset contains 300 binary features, represents the presence/absence of different keywords found in web pages. The goal is to classify whether a web page belongs to a certain category or not. The total sample size of the w8a training set and test set are respectively 49749 and 14951. As for data pre-processing, we remove features that include more than 49000 zeros on the training set and the number of remaining features is 156. Then we randomly split the training set into 40 machines evenly. Table 2 reports the average classification error on the test set of four distributed sparse estimators over ten replications. It can be seen that our proposed Pen-CASE estimator has the lowest classification error among four distributed sparse estimators.

Table 2: Classification errors and standard deviations of different estimators in w8a dataset.

Estimator	Naive	Pen-CASE	Debiased	5-step CSL	10-step CSL
Error	0.1082	0.1079	0.2239	0.1791	0.1789
SD	0.0000	0.0003	0.0452	0.0308	0.0307

References

Anil, R., Pereyra, G., Passos, A., Ormandi, R., Dahl, G. E., and Hinton, G. E. (2018). Large scale distributed neural network training through online distillation. In *International Conference on Learning Representations*.

Assran, M., Loizou, N., Ballas, N., and Rabbat, M. (2019). Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, volume 97, pages 344–353.

Battey, H., Fan, J., Liu, H., Lu, J., and Zhu, Z. (2018). Distributed testing and estimation under sparse high dimensional models. *Annals of statistics*, 46(3):1352—-1382.

Chen, L. and Zhou, Y. (2020). Quantile regression in big data: A divide and conquer based strategy. *Computational Statistics & Data Analysis*, 144:106892.

Chen, X., Liu, W., Mao, X., and Yang, Z. (2020a). Distributed high-dimensional regression under a quantile loss function. *Journal of Machine Learning Research (to appear)*.

Chen, X., Liu, W., Mao, X., and Yang, Z. (2020b). Distributed high-dimensional regression under a quantile loss function. *Journal of Machine Learning Research*, 21(182):1–43.

Chen, X., Liu, W., Zhang, Y., et al. (2019). Quantile regression under memory constraint. *The Annals of Statistics*, 47(6):3244–3273.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., and Ng, A. Y. (2012). Large scale distributed deep networks. In *Advances in Neural Information Processing Systems 25*, pages 1223–1231.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, 32(2):407–499.

Fan, J., Guo, Y., and Wang, K. (2019a). Communication-efficient accurate statistical estimation. *arXiv preprint arXiv:1906.04870*.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Fan, J., Wang, D., Wang, K., and Zhu, Z. (2019b). Distributed estimation of principal eigenspaces. *The Annals of Statistics*, 47(6):3009–3031.

Garber, D., Shamir, O., and Srebro, N. (2017). Communication-efficient algorithms for distributed stochastic principal component analysis. In *International Conference on Machine Learning*, pages 1203–1212.

Han, Y., Mukherjee, P., Ozgur, A., and Weissman, T. (2018). Distributed statistical estimation of high-dimensional and nonparametric distributions. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 506–510.

Javanmard, A. and Montanari, A. (2014). Confidence intervals and hypothesis testing for high-dimensional regression. *Journal of Machine Learning Research*, 15(82):2869–2909.

Jordan, M. I., Lee, J. D., and Yang, Y. (2019). Communication-efficient distributed statistical inference. *Journal of the American Statistical Association*, 114(526):668–681.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, J. D., Liu, Q., Sun, Y., and Taylor, J. E. (2017). Communication-efficient sparse regression. *Journal of Machine Learning Research*, 18(5):1–30.

Li, G., Peng, H., and Zhu, L. (2011). Nonconcave penalized M-estimation with a diverging number of parameters. *Statistica Sinica*, 21(1):391–419.

Lian, H. and Fan, Z. (2018). Divide-and-conquer for debiased l1-norm support vector machine in ultra-high dimensions. *Journal of Machine Learning Research*, 18:1–26.

Lin, N. and Xi, R. (2011). Aggregated estimating equation estimation. *Statistics and Its Interface*, 4(1):73–83.

Mcdonald, R., Mohri, M., Silberman, N., Walker, D., and Mann, G. S. (2009). Efficient large-scale distributed training of conditional maximum entropy models. In *Advances in Neural Information Processing Systems 22*, pages 1231–1239.

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. *Microsoft Research Technical Report*.

Rosenblatt, J. D. and Nadler, B. (2016). On the optimality of averaging in distributed statistical learning. *Information and Inference*, 5(4):379–404.

Shamir, O., Srebro, N., and Zhang, T. (2014). Communication-efficient distributed optimization using an approximate newton-type method. In *International Conference on Machine Learning*, pages 1000–1008.

Shang, Z. and Cheng, G. (2017). Computational limits of a distributed algorithm for smoothing spline. *Journal of Machine Learning Research*, 18(108):1–37.

Tishbirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Van de Geer, S., Bühlmann, P., Ritov, Y., Dezeure, R., et al. (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, 42(3):1166–1202.

Wainwright, M. J. (2009). Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso). *IEEE transactions on information theory*, 55(5):2183–2202.

Wang, J., Kolar, M., Srebro, N., and Zhang, T. (2017a). Efficient distributed learning with sparsity. In Precup, D. and Teh, Y. W., editors, *International Conference on Machine Learning*, volume 70, pages 3636–3645.

Wang, J., Wang, W., and Srebro, N. (2017b). Memory and communication efficient distributed stochastic optimization with minibatch prox. In *Conference on Learning Theory*, pages 1882–1919.

Wang, X., Yang, Z., Chen, X., and Liu, W. (2019). Distributed inference for linear support vector machine. *Journal of Machine Learning Research*, 20:1–41.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38(2):894–942.

Zhang, Y., Duchi, J., and Wainwright, M. (2015). Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 16(102):3299–3340.

Zhang, Y., Duchi, J. C., and Wainwright, M. J. (2013). Communication-efficient algorithms for statistical optimization. *Journal of Machine Learning Research*, 14(68):3321–3363.

Zinkevich, M., Weimer, M., Li, L., and Smola, A. J. (2010). Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems 23*, pages 2595–2603.